Introduction

A security review of **Gondi V3** protocol was done by **qckhp**.

Disclaimer

A smart contract security review can never verify the complete absence of vulnerabilities. This is a time, resource and expertise bound effort where I try to find as many vulnerabilities as possible. I can not guarantee 100% security after the review or even if the review will find any problems with your smart contracts.

About Gondi V3

Gondi is a decentralized non-custodial NFT lending protocol engineered to enable the most capital efficient loan primitive and credit market for NFTs.

Severity classification

Severity	Impact: High	Impact: Medium	Impact: Low
Likelihood: High	Critical	High	Medium
Likelihood: Medium	High	Medium	Low
Likelihood: Low	Medium	Low	Low

Impact - the technical, economic and reputation damage of a successful attack

Likelihood - the chance that a particular vulnerability gets discovered and exploited

Severity - the overall criticality of the risk

Security Assessment Summary

review commit hash - f78f25f1cb72d472aa03ef7a42345f5e0da5480f

Findings Summary

The following number of issues were found, categorized by their severity:

- Critical: 0 issues
- High: 0 issues
- Medium: 2 issues
- Low: 1 issues

ID	Title	Severity
[M-01]	MultiSourceLoan incompatible with ERC-1271	Medium
[M-02]	OraclePoolOfferHandler confirmCollectionFactors() can be DOS'd	Medium
[L-01]	confirmBaseInterestAllocator can be front ran on first call	Low

Detailed Findings

[M-01] MultiSourceLoan incompatible with ERC-1271

Description

In the current MultiSourceLoan implementation `_validateOfferExecution()` function checks if `if (lender.code.length != 0) {` and if true, it's assuming the lender is the LoanManager contract, which is making the MultiSourceLoan contract incompatible with ERC-1271.

Recommendations

Add a check if the lender is indeed a registered loanManager.

Resolution

Fixed by commit [10d48b5]

[M-02] OraclePoolOfferHandler confirmCollectionFactors() can be DOS'd

Description

In `OraclePoolOfferHandler` the `confirmCollectionFactors` function can be DOS'd making it impossible to update the CollectionFactors. By sending empty arrays as input parameters the value of `getProposedCollectionFactorsSetTs` is updated to `type(uint256).max`, without updating any other values.

Recommendations

Add a minimum length check for the inputs.

Resolution

Team will fix the issue.

[L-01] confirmBaseInterestAllocator can be front ran on first call

Description

The `Pool` contracts `confirmBaseInterestAllocator()` function has to be called by the deployer before transfering any funds to the contract!

Recommendations

Need make sure to call the `confirmBaseInterestAllocator()` during deployment, or add `onlyOwner` modifier.

Resolution

Team will fix the issue.